Lars Borin
Centrum för datorlingvistik
Sturegatan 13B[V]
752 23 UPPSALA

ETT TEXTDATABASSYSTEM FÖR LINGVISTER

1 Bakgrund

Under en längre tid har Centrum för datorlingvistik utvecklat och tillhandahållit en rad programpaket för textbearbetning i olika former. Dessa programpaket har använts och används under samlingsnamnet TEXTPACK (Nurmi & Rosén 1983; Rosén 1983; Rosén & Sjöberg 1983) av språkvetare från olika språkinstitutioner, främst vid Uppsala Universitet.

Till de språk som i skrivande stund datorbearbetas med de olika TEXTPACK-rutinerna hör bl.a. engelska, finska, persiska, ryska, sanskrit och svenska, med större eller mindre problem vad gäller inmatning, teckenrepresentation, kollatering och utskrift.

De dataformat TEXTPACKsystemet arbetar med är i hög grad standardiserade, men de accessmetoder det använder är avhängiga av det operativsystem vi arbetar under (IBM MVS, se (PL181b)), och de accessoperatorer programmeringsspråket tillhandahåller (PL/1, se (PL181a)). För att komma åt detta maskinberoende har vi vid Centrum för datorlingvistik under ca. två och ett halvt års tid sysslat med att utveckla ett mer generellt textbearbetningssystem.

2 Förutsättningar och problem

Vid utvecklingsarbetet har vi utgått ifrån att datoranvändarna vid universiteten av ekonomiska skäl under ganska lång tid framåt kommer att vara hänvisade till de typer av datorer och in- och utmatningsutrustning vi har idag, med de problem detta innebär för den språkvetenskapliga databehandlingen. Några sådana problem är:

Vid inmatning kommer den normala utrustningen att bestå av en vanlig bildskärmsterminal med svenskt tangentbord[1]. För de västeuropeiska språkens del kan man räkna med att i högre grad än hittills kunna utnyttja sättband från datorsättningsutrustningar[2], men när det gäller andra språk, kan möjligheterna i det avseendet vara betydligt mer begränsade av olika skäl (låg datoriseringsgrad, trög byråkrati osv.).

Ett område där utvecklingen går snabbt är optisk läsning, och där kan

Ett textdatabassystem för lingvister
Lars Borin
Proceedings of NODALIDA 1983, pages 37-47

man säkert förvänta sig en hel del intressanta nyheter så småningom.

När det gäller presentationen av det språkliga materialet ser bilden ljusare ut, åtminstone vad gäller utskrift på papper. De skrivare man idag använder vid stora datoranläggningar (mest radskrivare) är visserligen rätt begränsade vad gäller teckenvariation, något som också gäller de flesta skönskrivare för ordbehandling, men nu börjar de s.k. laserskrivarna bli allt billigare och utgör idag ett ekonomiskt bärkraftigt alternativ till de traditionella skrivarna[3]. Med laserskrivarna har man i princip inte längre några begränsningar vad gäller teckenuppsättningar vid utskriften, och de skriver dessutom med en kvalitet som lämpar sig för publicering. Handlar det däremot om utskrift på bildskärm gäller vad som sagts om bildskärmsterminaler ovan.

Lagringen och bearbetningen är de kritiska stegen i den datoriserade textbehandlingen; inmatnings- och presentationsproblem är mest en fråga om att ha ett lämpligt gränssnitt mot den lagrade informationen. Att finna en lämplig form för lagring och bearbetning av text ur lingvistisk synvinkel hänger intimt samman med datorns teckenrepresentation.

Den enda mer vittgående förändring man kan förutspå för språkvetarnas del, är en ytterligare utbyggnad av den distribuerade databehandlingen, i och med att fler och fler universitetsinstitutioner, även språkvetenskapliga, skaffar sig mikrodatorer. Dessa har ju blivit allt billigare och kraftfullare sedan de först introducerades, och den utvecklingen kan förväntas fortgå ett tag till. Av den anledningen förutser vi att efterfrågan på portabel (maskinoberoende) programvara kommer att öka i framtiden.

Med dessa förutsättningar i åtanke siktar vi på att utveckla ett generellt system för lingvistisk textbearbetning.

3 Krav

Följande krav vill vi att detta system skall uppfylla:

1) Det skall vara **användarvänligt.** Man vill i så hög grad som möjligt automatisera de administrativa uppgifter användaren alltför ofta själv tvingas utföra, såsom att se till att de data systemet arbetar med alltid är konsistenta inbördes. Ju fler sådana rutiner systemet kan klara av, i desto högre grad kan man förskjuta användarens aktivitet in på de lingvistiskt relevanta problemens område. Om man ser det hela ur en annan synvinkel kan man säga, att en stor del av all den språkvetenskapliga databehandling som görs idag består av rent 'mekanisk' textbearbetning (grafordsstatistik, produktion av konkordanser, etc.), och att det är angeläget att göra sådana bearbetningar så enkla som möjligt att utföra för användaren.

2) Ett annat krav är att systemet skall vara i möjligaste mån **generellt,**

med en inbyggd beredskap att klara av de mest skilda särspråkliga problem som kan tänkas dyka upp.

En aspekt på generalitetsproblemet är frågan om vilken sorts information olika användare vill lagra om sina texter och textelement; ett idealiskt system bör ha en flexibel informationsstruktur, där det är lätt att finna plats för nya typer av information, och där man har förutsett att det kan komma att läggas till systemrutiner som bearbetar denna nya information. Systemet måste alltså vara öppet, både vad gäller informationsstruktur och mjukvara.

3) Systemet bör vara i mesta möjliga mån **portabelt.** Detta hänger ihop med den utveckling mot distribuerad databehandling vi förutsåg ovan; i och med att antalet datorer ökar, kommer också efterfrågan på portabel programvara att öka. Om man från början gör programmen portabla minskar man dessutom den möda som måste läggas ned på konvertering och eventuell omprogrammering.

4) Som nämndes i avsnitt 2 ovan, är den teckenrepresentation man använder helt avgörande för hur man kan lagra och bearbeta språkligt material. Det man önskar sig av teckenrepresentationen är följande:

1. Att varje tecken skall vara unikt.
2. Att ens tecken sorteras (kollateras) på det sätt man själv bestämmer.
3. Ordlängder måste stämma med språkets ortografi; digram, trigram osv. skall kunna räknas som en bokstav.
4. Att i vissa fall kunna behandla tecknen på (allo-)grafnivå ("Appearance" i Xerox83 (s. 4)), och i andra fall bara räkna med grafemen, tecknens "kanoniska" identitet.

5) Eftersom många tecken, främst skiljetecken, inte är entydiga, jfr. förkortningspunkt och meningsavslutande punkt, vill man kunna betrakta sådana flertydiga tecken som homografer, och få dem markerade som sådana i texten, för att man sedan skall kunna ta hand om dem med någon form av homografseparering.

3.1 Ungerskan som exempelproblem

För att i någon mån konkretisera de problem man kan stöta på vid språklig databehandling, valde jag ungerskan som typfall, mest av en slump (jag höll på att studera ungerska då jag började arbeta på det textbearbetningssystem som redovisas i nästa avsnitt), men den visade sig vara mycket intressant i detta sammanhang.

Ungerskan skrivs med ett modifierat latinskt alfabete, enligt följande:

(a á) b c cs d dz dzs (e é) f g gy h (i í) j k l ly m
n ny (o ó) (ö ő) p r s sz t ty (u ú) (ü ű) v z zs

Man lägger märke till den stora mängden digram (och trigrammet 'dzs'), som i sorteringshänseende uppför sig som enkla bokstäver, något som datorn inte är förberedd för, då den sorterar tecken för tecken. Långa vokaler skrivs med en akut accent över vokaltecknet (utom långt /ö/ och/ü/ som skrivs med två akuta accenter) och samsorteras med motsvarande korta vokal i ordböcker o.dyl. (markerat med parenteser ovan), men i övrigt betraktas som egna bokstäver, medan långa konsonanter dubbeltecknas, och betraktas som två separata bokstäver. En egenhet i den ungerska ortografin är dock att för de långa varianterna av de konsonanter som skrivs med digram eller trigram fördubblas endast det första tecknet (långt /gy/ skrivs <ggy>), men kombinationen skall sorteras som om den bestod av två separata bokstäver (<ggy> skall sorteras som <gygy>). Versaler samsorteras med gemena, men som i andra språk med både stora och små bokstäver, har de en inbördes sorteringsordning för sådana ord som skiljer sig åt enbart med avseende på dessa (som svenska 'sten' och 'Sten' (namnet)). Se vidare MHSz79.

Den ungerska ortografin orsakar ett annat problem, som egentligen inte hör hemma här, men som det ändå kan vara illustrativt att ta upp i det här sammanhanget: Det kan inträffa i en morfemgräns, att två tecken som ingår i ett bokstavsdigram kommer att stå intill varandra, utan att de för den skull utgör någon bokstav; orden **kőzség** 'kommun' och **kőzsák** 'stenpelare, rauk' delas upp i bokstäver som följer:

<k> <ö> <z> <s> <é> <g> - <k> <ő> <zs> <á> <k>[4].

Här följer ett exempel på hur man skulle sortera några ord enligt de ungerska ortografiska normerna, och hur ett normalt datorsorteringsprogram skulle sortera samma ord (vi förutsätter ett ungerskt tangentbord, för att kunna bortse från problemet med att representera de långa vokalerna[5]):

Ungersk sortering		Datorsortering
kása	'gröt'	kassza
kastély	'lustslott'	kastély
kasza	'lie'	kasza
kaszinó	'kasino'	kaszinó
kassza	'kassa'	kaszt
kaszt	'kast (samhällsklass)'	kása

Datorsorteringen går uppenbarligen inte att använda utan modifikationer.

Kommersiellt tillgängliga sorteringsprogram (ex.-vis SYNCSORT (SyncS80), som finns tillgängligt vid UDAC, där vi gör många av våra bearbetningar) brukar tillåta samsorteringar och ändringar av sorteringsordningen, men man brukar inte kunna komma åt problemet med bokstavsdigram (och längre "-gram") inom ramarna för dessa.

4 En lösning: Textdatabassystemet

Det stod ganska snart klart för oss, att ett system uppbyggt kring en databas vore ett bra sätt att uppfylla de krav och lösa de problem som jag redogjort för i de föregående två avsnitten. Med ett lämpligt valt (läs: portabelt) databassystem, får man flera av kraven uppfyllda nästan gratis:

- Man vinner en hel del i fråga om konsistens jämfört med vanliga, fil-orienterade bearbetningar: Med riktigt skrivna access- och uppdaterings-rutiner ser man till att ändringar i den lagrade informationen slår igenom överallt där det behövs, utan att man som användare behöver hålla reda på vilka filer man har ändrat i, och vilka som måste ändras som en konsekvens av detta.

- Man kan skriva en mer integrerad mjukvara: Eftersom allting händer i en och samma databasrymd, där all information är lagrad i ett enhetligt format, har man i varje ögonblick tillgång till denna information, som man annars skulle behöva hoppa mellan flera olika undersystem (specialskrivna program, operativ-systemrutiner, terminalsystemrutiner osv.) för att komma åt, något som inte kan göra annat än stjäla tid från ens lingvistiska problem, textbearbetningen.

- Man får en i hög grad portabel mjukvara: Genom att man arbetar med databas-systemets accessoperatorer, som förblir desamma oberoende av vilken dator man använder, så kan man flytta sina textbearbetningsrutiner mellan olika datorer utan att behöva skriva om dem.

Vårt textdatabassystem under utveckling består av två delar:

1) Ett generellt relationsdatabassystem, Mimer/DB (Mimer 1982a), som tillhandahåller rutiner för att skapa, ladda, ändra och tömma databanker, och som tar hand om alla operativsystemberoende dataförflyttningar och ersätter dessa med egna databasaccessrutiner, lika på alla datorer som har Mimer/DB[6]. Detta innebär naturligtvis att de program som använder Mimer/DB inte behöver skrivas om för att kunna flyttas mellan olika datorer.

2) En accessmetod, specialdesignad för att lagra och bearbeta text ur lingvistisk synvinkel, som vi själva utvecklat. I första hand har målet varit att implementera de funktioner som TEXTPACK tillhandahåller (se nedan), men det är meningen att man fortsättningsvis skall kunna utnyttja de speciella fördelar databasorganisationen ger för att kunna underlätta användarens arbete.

Databassystemet tillhandahåller ett "virtuellt" filhanteringssystem, och vår accessmetod ett slags operativsystem, speciellt ägnat för lingvistisk bearbetning av textmaterial. Båda dessa komponenter är i hög grad portabla. De textbearbetningsfunktioner som implementerats är följande:

- Möjlighet att definiera alfabeten/skriftsystem med upp till 65 535 (2^{16}-1) tecken i en användarbestämd sorteringsordning, samt att definiera textformatterande specialtecken och teckenkombinationer som skall specialbehandlas i kollateringshänseende.

- Inläsning av texter i en databank, där dessa lagras i ett kompakt internt format.

- Utskrift, på bildskärm eller papper, av grafordslistor, meningslistor och grafordskonkordanser.

Under implementering är:

- Rutiner för datorstödd lemmatisering och homografseparering av texter i databasen.

- Rutiner för att ändra den information som finns i databasen, speciellt för rättning av texter och ändring av alfabetesdefinitioner. Här kan man inte använda ett konventionellt databasfrågespråk (som Mimer/QL (Mimer 1982b)), då det interna format texterna lagras i kräver en rätt omfattande avkodning innan det blir läsbart. Dessutom måste man se till att all information som hör ihop förblir konsistent.

- Rutiner för att hämta in och lagra information om de texter som lagras i databasen, ss. texttyp, genre, författare osv..

Det användardefinierade alfabetet har följande egenskaper:

- Varje tecken har en unik 16-bitsrepresentation[7],

- som samtidigt utgör dess grundposition i sorteringsordningen. Dessutom kan man för varje tecken ange en samsorteringsposition, eller tala om att ett tecken (t.ex. bindestreck) skall ignoreras vid sorteringen. Allt detta görs vid alfabetesdefinitionen för den egna databasen.

- Digram, trigram osv. som räknas som egna bokstäver enligt den aktuella ortografin behandlas som odelbara enheter i den interna representationen (de tilldelas en egen 16-bitskod).

- Varje teckens 'utseende' lagras separat från själva grundtecknet, detta för att användaren skall ha största möjliga frihet när det gäller sådana saker som att kunna ta hänsyn till versaler eller ignorera dem vid olika slags bearbetningar eller olika bearbetningsfaser. För varje tecken har man (arbiträrt) möjlighet att definiera upp till fem gemena (ex.: grekiskt sigma har två varianter) och fem versala allografer; bokstavsdigrammen brukar ha två allografer, t.ex: 'Cs', 'CS' (/č/ i ungersk ortografi; se avsnitt 3 ovan).

Texternas interna format är delvis dikterat av lingvistiska hänsyn och delvis av praktiska begränsningar i MIMER/DB.

När en text läses in i databasen går den först igenom en konverteringsrutin som slår upp varje tecken i ett bokstavsträd. Bokstavsträdet innehåller information om hur tecknen skall sorteras, vad tecknen har för typ (alfabetiskt, siffra, skiljetecken, specialtecken osv.). Tecknen konverteras till det ovan nämnda 16-bits teckensetet och sparas undan i grafordssträngar. Grafordssträngarna i sin tur läggs in i en strängtabell, och representeras sedan i databasens andra tabeller som pekare till denna, detta på grund av att det använda databassystemet inte tillåter variabel fältlängd i tabellerna. Av utrymmesskäl vill man inte överallt reservera plats för det längsta ord man kan tänkas stöta på, eftersom databasen då till stor del skulle bestå av tomrum.[8]

5 Framtiden

Sammanfattningsvis kan man säga att vi strävar efter att utveckla ett flexibelt och portabelt system för lingvistisk textbearbetning, byggt på en databas, för

att därigenom kunna ge systemet mesta möjliga kontroll över datasambanden i den information som lagras.

Textdatabassystemet står ännu på experimentstadiet; man kan läsa in texter, och man kan skriva ut dem på olika sätt, men man har ännu ingen möjlighet att ändra i dem. Vi arbetar dock kontinuerligt med att utveckla och förfina detta redskap för lingvistiskt orienterad textbearbetning, och räknar med att det i framtiden kommer att bli vårt standardverktyg för sådana ändamål.

För det framtida utvecklingsarbetet på textdatabasen svarar i första hand Valentina Rosén vid Centrum för datorlingvistik.

Noter

1. Dock har det gjorts enstaka ansträngningar att konstruera inmatningsutrustning för specifika skriftsystem, ss. Nancarrows "Ideo-Matic Encoder" (Nancarrow 1980; 1981; 1982).

2. Språkdata i Göteborg är något av pionjärer i Sverige när det gäller att utnyttja sättband. Bland annat är Nusvensk frekvensordbok (Allén 1970; 1971; Allén et al. 1975; 1980) baserad på material framtaget från tidningssättband.

3. I BYTE:s marsnummer för 1984 kan man läsa om en ny japansk laserskrivarmekanism som säljs för 1000-2000 dollar (Willis 1984). Om man lägger till kostnaden för kontrollelektroniken, hamnar en färdig laserskrivare i samma prisklass som de dyrare typhjulsskrivarna.

4. Problem av denna typ får snarast lösas genom lemmatisering. Datorn har ingen möjlighet att lösa dem utan lingvistisk kunskap, och när man börjar laborera med sådan, har man redan förflyttat sig bort från den rent 'mekaniska' textbearbetningen, som Rolf Gavare mycket riktigt påpekade efter den muntliga presentationen.

5. Ungerska skrivmaskiner har separata tangenter för de långa vokalerna (á, é osv.), men inte för de bokstäver som skrivs med mer än ett tecken.

6. Mimer finns f.n. (våren 1984) implementerat på ett tjugotal dator - operativsystemkombinationer, däribland flera mikro- och minidatorer (Mimer 1983).

7. Alfabeten med maximalt 255 tecken får rutinmässigt en 8-bitsrepresentation för att man skall spara utrymme i databasen.

8. Rolf Gavare redogör i sin lilla skrift "Lexikografisk alfabetisering" (Gavare 1983) för många av de problem man har att brottas med vid sortering av språkligt material i datamaskinella sammanhang, och presenterar en sorteringsalgoritm som tar hänsyn till de faktorer jag har redogjort för, och en del andra dessutom. Hans resonemang gäller främst svenskan, men hans algoritm är mycket genomtänkt, och t.ex. hans sätt att behandla diakritika som inte är en integrerad del av en bokstav borde passas in någonstans i vårt databassystem.

Referenser

Allén, S. 1970. Nusvensk frekvensordbok baserad på tidningstext. 1. Graford, homografkomponenter. Stockholm.

---------- 1971. Nusvensk frekvensordbok baserad på tidningstext. 2. Lemman. Stockholm.

Allén, S. et al. 1975. Nusvensk frekvensordbok baserad på tidningstext. 3. Ordförbindelser. Stockholm.

Allén, S., S. Berg, J. Järborg, J. Löfström, B. Ralph, C. Sjögreen 1980. Nusvensk frekvensordbok baserad på tidningstext. 4. Ordled, betydelser. Stockholm.

Borin, L. 1983. Systemdokumentation för textdatabasen: Version 1.0. UCDL-R-83-3. Uppsala.

Teletex80 = Character Repertoire and Coded Character Sets for the International Teletex Service. CCITT Recommendation S.61. Geneva 1980.

Gavare, R. 1983. Lexikografisk alfabetisering. Rapporter från Språkdata 14. Göteborg.

JIS78 = JIS (Japanese Industrial Standard) C6226-1978. Code of the Japanese Graphic Character Set for Information Interchange. Japanese Standards Association. Tokyo 1978.

MHSz79 = A magyar helyesírás szabályai (De ungerska rättskrivningsreglerna). Tizedik kiadás. Budapest 1979.

Mimer 1982a. Mimer/DB Reference Manual. Mimer Information Systems. Uppsala

----- 1982b. Mimer/QL Reference Manual. Mimer Information Systems. Uppsala.

----- 1983. Mimer Newsletter, Nr. 1, Januari 1983.
Uppsala Datacentral. Uppsala.

Nancarrow, P. H. 1980. A System for Processing Tibetan Texts in their Original Orthography. i: ALLC Journal, 1980, no. 1.

---------- 1981. A Brief Account of the Development and First Major Installation of the Ideo-Matic Chinese Character Encoder. i: ALLC Bulletin, 1981, no. 3.

---------- 1982. Processing of Ancient Egyptian Hieroglyphic Texts by Computer. i: Richard W. Bailey (ed.): Computing in the Humanities. Amsterdam - New York - Oxford.

Nurmi, S. & V. Rosén 1983. TEXTPACK - Basprogram för bearbetning av text (Del 1). Centrum för datorlingvistik. Uppsala.

PL181a = OS and DOS PL/1 Language Reference Manual. GC26-3977-0. IBM 1981.

PL181b = OS PL/1 Optimizing Compiler: Programmer's Guide. SC33-0006-5. IBM 1981.

Rosén, V. 1983. TEXTPACK - Stegen "PRE" och "BAS" (Del 2). Centrum för datorlingvistik. Uppsala.

Rosén, V. & M. Sjöberg 1983. TEXTPACK - Stegen "RAD" och "MEN" (Del 3). Centrum för datorlingvistik. Uppsala.

SyncS80 = A Programmer's Guide to SyncSort: OS & OS/VS. WCS-0101-0. Whitlow Computer Systems 1980.

Willis, R. 1984. The Japan Shows. An Update on the Japanese Computing Scene. BYTE, Vol. 9, No. 3, March 1984.

Xerox83 = Xerox System Integration Standard XSIS 058305. Character Encoding Standard, May 1983. El Segundo, California 1983.